

I. modul: Kombinációs hálózatok

4. lecke: Házárdok kombinációs hálózatokban; a multiplexer, mint programozható logikai hálózat

Cél:

A lecke célja, hogy a hallgató megismerje a kombinációs hálózatokban előfordulható házárdok okait és kiküszöbölésük módjait. Értelmezze a multiplexerek, mint kombinációs hálózati elemek működését, és programozható logikai hálózatként alkalmazni tudja őket a digitális hálózatokban.

Követelmények:

Ön akkor sajátította el megfelelően a tananyagot, ha

- ismeri a házárdok típusait és kiküszöbölésük módjait
- alkalmazni tudja a multiplexereket programozott logikai hálózati egységként

Időszükséglet:

A tananyag elsajátításához kb. 60 percre van szükség.

Kulcsfogalmak:

- kapu-késleltetési idő, tranziens idő
- statikus, dinamikus, funkcionális házárd
- adatút kijelölő eszköz

Tevékenység: tanulmányozza a leckében található példán keresztül a statikus hazard felismerését Karnaugh táblás ábrázolásmód alapján. Igazságtábla felrajzolásával vagy mintermes kanonikus függvényalak meghatározásával adja meg két változóra nézve az AND, NAND, OR, NOR függvényrelációk programozott multiplexerrel történő realizációját szimbólum szintű sematikus ábrák segítségével.

4.1 Hazardok

A kombinációs hálózatok legfőbb ismérve - a tanultak alapján - a következők szerint foglalható össze: egy kombinációs hálózat viselkedésének legfontosabb sajátossága, hogy egy meghatározott bemeneti kombináció ismételt rákapcsolásaira - a tranziens idő eltelte után - mindig ugyanazt a kimeneti kombinációt szolgáltatja, függetlenül attól, hogy az adott bemeneti kombináció két rákapcsolása között milyen más bemeneti kombinációkat kapcsoltunk a hálózatra.

Az eddigiek során a kombinációs hálózatokat statikusan szemléltük, nem foglalkoztunk a tranziensekkel, azaz az átmeneti jelenségekkel. Általában megköveteljük, hogy a bemeneti változások hatására a kimenet a specifikációnak megfelelően reagáljon. Ez azonban a gyakorlatban, vagyis az áramkörü realizációk esetében nem mindig garantálható. Ezeket a jelenségeket, vagyis a specifikációtól való eltéréseket hazardoknak nevezzük.

4.1.1 Statikus hazard

4.1.1.1 A statikus hazard meghatározása

Statikus hazardról akkor beszélünk, ha egyetlen bemeneti változó logikai értékének megváltozásakor a kimenet a specifikáció szerint nem változna, de a realizált hálózat kimenetén mégis átmeneti változás zajlik le.

Ez az átmeneti értékváltozás –a kimenő logikai változó(k) lehetséges bináris értékeiből kiindulva- kétféle statikus hazardot eredményezhet, melynek alapján a két hazardtípus a következő:

- *'0'-ás típusú statikus hazard*
- *'1'-es típusú statikus hazard*

A '0'-ás típusú statikus hazardról akkor beszélünk, ha a specifikált hálózat kimenete a bemeneti változás ellenére alacsony logikai szinten ('0') kell, hogy maradjon, de a realizált hálózat - átmenetileg - egy magas, '1'-es logikai szintű impulzust mutat.

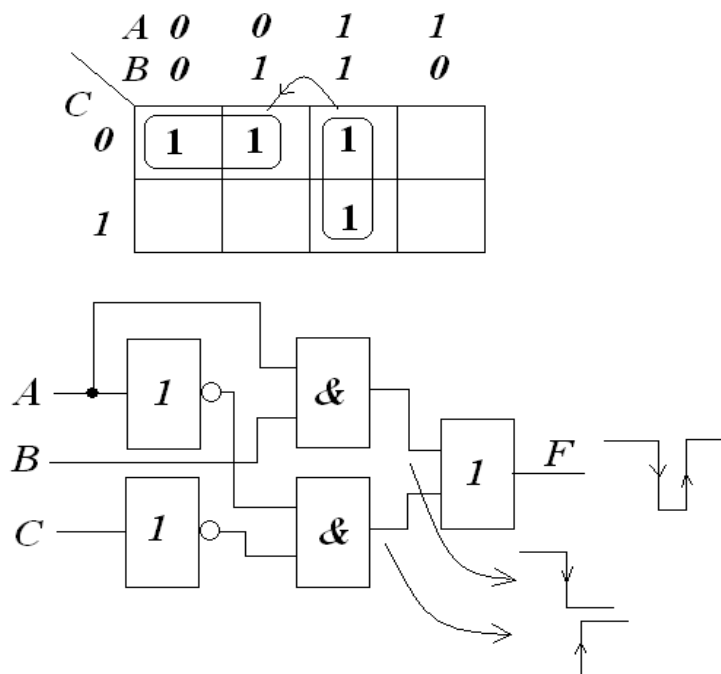
'1'-es típusú statikus hazárdról pedig akkor beszélünk, ha a specifikált hálózat kimenete a bemeneti változás ellenére magas logikai szinten ('1') kell, hogy maradjon, de a realizált hálózat - átmenetileg - egy alacsony, '0'-ás logikai szintű impulzust mutat.

4.1.1.2 Statikus hazárdok kiküszöbölése

A statikus hazárdok megszüntethetők ún. redundáns implikánsok bevezetésével.

Példa:

a 4.1.1.2.a ábrán láthatóak egy Karnaugh táblás ábrázolással definiált hárombemenetű és egykimenetű kombinációs hálózat függvény prímisszimplikánsai, valamint az ehhez tartozó egyszerű kapu implementáció. Az eddigi tanulmányaink alapján kijelenthető, hogy a két prímisszimplikáns ($\bar{A}\bar{C}$, AB) összegszerű felírásával megadható az hálózat viselkedésének logikailag korrekt leírása.



4.1.1.2.a ábra[1]

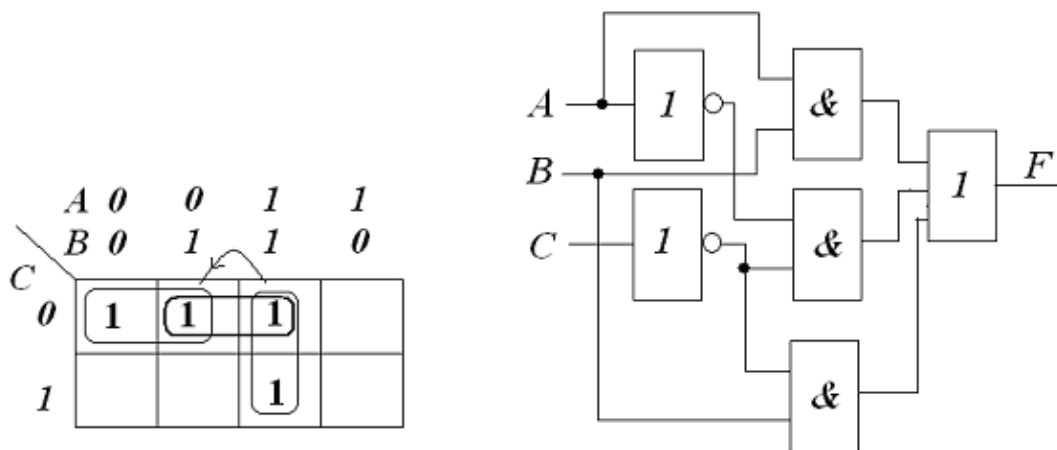
Egy 'A,B,C', bemenetű és 'F' kimenetű hálózat Karnaugh táblás és kapu szintű specifikációja, '1'-es típusú statikus hazárddal

Vizsgáljuk most meg, hogy mi történik pl. abban az esetben, ha az ABC bemeneteken az $1\ 1\ 0 \rightarrow 0\ 1\ 0$ bitkombináció változás végbemegy, vagyis azt az átmenetet, amikor az A változó értéke magas szintről alacsony szintre vált.

A tranziens analízis során feltételezzük, hogy minden egyes kapufokozatnak késleltetési ideje van. Beláthatjuk, hogy az A 'lefutásakor' a VAGY kapu egyik bemenetén a magas szint biztosan előbb fut '0'-ra, mint a másik '1'-re, hiszen ez utóbbi változás két kapun, egy inverteren és egy ÉS kapun halad át, míg az előbbi késleltetése csak egy ÉS kapunyi. Van tehát egy átmeneti idő-intervallum, amikor a VAGY kapu egyik bemenete már nem, a másik bemenete még nem '1'-es szintű. Annak ellenére tehát, hogy a kimenetnek a logikai specifikáció szerint magas szinten kéne maradnia, átmenetileg lefut '0'-ra. Ez a rövid idejű '0' impulzus egyrészt nem felel meg a logikai specifikációnak, másrészt egyértelmű, hogy a kapuk fizikai késleltetése okozza azt.

Ebben az esetben megoldást jelent egy ún. átfedő, vagy redundáns prímimplikáns felírása illetve alkalmazása (4.1.1.2.b ábra). Az '1'-es típusú statikus hazard veszély mindig abból adódik, hogy különböző prímimplikánsokhoz tartozó '1'-es mintermek kerülnek egymás mellé. Az áthidaló implikáns ($B\bar{C}$) bevezetésével elérjük, hogy az átmenet alatt a kimenet logikai szintje állandó maradjon. A redundáns, ugyanakkor hazard-mentesítő implikánssal együtt felírt kifejezés:

$$F(A, B, C) = \bar{A}\bar{C} + AB + B\bar{C}$$



4.1.1.2.b ábra[1]

Statikus hazardmentesített hálózat

4.1.2 Dinamikus hazard

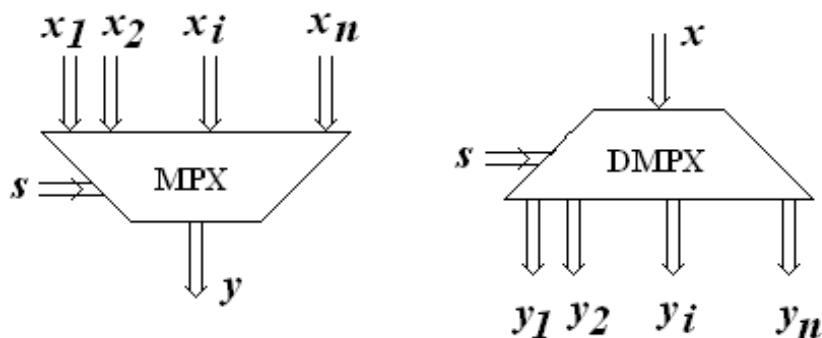
Dinamikus hazardról beszélünk, ha egy bemeneti-változó értékváltására a kimenetnek logikai értéket kell váltania, de ez egy átmeneti visszatérés kíséretében zajlik le. Belátható, hogy a dinamikus hazard többszintű hálózatokban lép fel akkor, ha a hálózat valamely része nincs statikus hazardoktól mentesítve. A statikus hazardokat kell kiküszöbölni, így a dinamikus hazard eltűnik.

4.1.3 Funkcionális hazard

Funkcionális hazard esetén több bemeneti változó együttes változása a kimeneten nem előírászerű, többszörös szintváltást eredményezhet. Ezek a hazardok csak késleltetési manipulációkkal küszöbölhetők ki, de az a legjobb, ha a tranziensek továbbterjedését szinkronizációval megakadályozzuk.

4.2 Programozható kombinációs hálózatok

A digitális architektúrákban alkalmazott kombinációs hálózatok egy speciális családját alkotják a multiplexerek (MPX vagy MUX) és a demultiplexerek (DMPX vagy DEMUX). Ezek lényegében az összetett digitális egységek azon csoportjába sorolhatók, melyek adatút kijelölő egységekként funkcionálnak. Általános felépítésüket a 4.2.a ábrán tanulmányozhatjuk:



4.2.a ábra[1]

A multiplexerek(baloldali kép) és demultiplexerek(jobboldali kép) szimbólumai

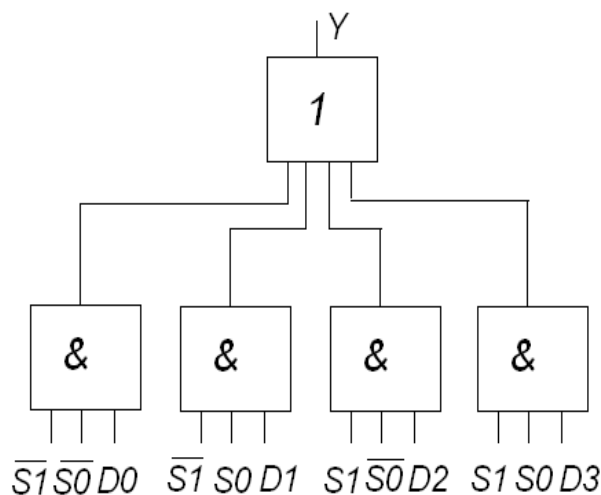
Az adatutak kijelölése vezérlőbemenetek segítségével (S), címzéssel történik. Multiplexereknél több forrás közül kijelöljük azt (x_i), amelynek adata a kimenetre (y) kerül, míg a demultiplexerek esetén azt a kimenetet (y_i) címezzük meg, amelyen az egyetlen forrás adatát (x) meg kívánjuk jeleníteni. A címzés általában bináris kóddal történik, így eleve kizárt az ellentmondásos kettős címzés.

Ebben a modulrészben a multiplexerek felépítésével foglalkozunk részletesebben, mivel a következő fejezetben a szinkron sorrendi hálózatok tervezésekor, mint programozható logikai hálózati elemet használjuk majd fel egy speciális célarchitektúra alkotóelemeként.

4.2.1 A multiplexerek általános felépítése

Példa:

vizsgáljuk meg egy négybemenetű, egykimenetű multiplexer (MPX4-1) felépítését a 4.2.1.a ábra segítségével:



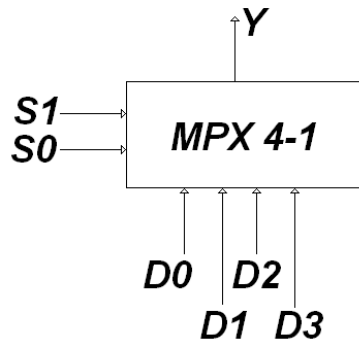
4.2.1.a ábra[1]

Egy MPX4-1 multiplexer kapuszintű felépítése

Ez a multiplexer négy darab 3-bemenetű 'ÉS' kapuból, és egy darab 4-bemenetű 'VAGY' kapuból, illetve az ezeknek megfelelő bemenetszámú 'NEM-ÉS' kapukból áll. (A kiválasztó bemenetek negált szintjei inverterekkel vannak megvalósítva.)

Természetesen a realizáció a gyakorlatban NAND-NAND kivitelben is létezik. Belátható, hogy ezek az eszközök valóban kombinációs hálózatként épülnek fel.

A példánkban szereplő multiplexer közepes integráltságú logikai áramkörökben (MSI= Medium-Scale Integration) használatos jelölése a 4.2.1.b ábrán látható:



4.2.1.b ábra[1]

MPX4-1 multiplexer MSI szimbóluma

4.2.2 A multiplexer, mint programozható logikai hálózat

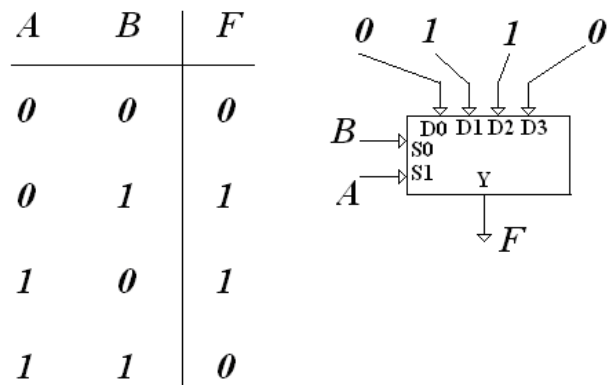
A multiplexerek - adatút választó funkciójuk mellett – az AND-OR felépítésű struktúrájukból fakadóan alkalmasak egyszerűbb logikai függvények kapusztintű megvalósítására is. Ezzel lényegében ún. programozott logikai hálózatként működnek. Ennek lényegét a következők szerint lehet értelmezni: a mintermes kanonikus alakban megadott függvény mintermjait a címző (vezérlő) bemenetekre adott címek képviselik, és a megcímzett adatbemenetre rá kell kapcsolnunk az adott mintermhez tartozó logikai értéket. Ezeknek a logikai konstansoknak a bemenetekre való kapcsolását tekinthetjük a multiplexerek programozásának

Példa:

valósítsuk meg két változó (A, B) esetében az EXOR függvényt (F) multiplexer felprogramozásával!

Ebben az esetben – a fenti meghatározás szerint – az F függvény A, B változói adják a multiplexerünk két címző bemenetét (4.2.2.a ábra). Ez azt is jelenti egyben, hogy a szóban forgó függvény realizálásához egy MPX4-1-es multiplexert kell felprogramoznunk. Az EXOR relációt leíró függvényalak (F) két minterme ($\bar{A}B, A\bar{B}$) a multiplexer D_1 és D_2 adatbemenetét címzi meg, és ezekre az adatbemenetekre kell rákapcsolni konstans értékként a mintermhez rendelt '1'-es logikai értéket:

$$F = \bar{A}B + A\bar{B}$$



4.2.2.a ábra[1]

EXOR függvény(F) megvalósítása MPX4-1 multiplexerrel

Felhasznált irodalom:

[1] Dr. Keresztes Péter: Digitális hálózatok (HEFOP elektronikus Jegyzet)